協調型仮想計算機システムにおける協調機構

荒木 裕靖 †
† 立命館大学大学院理工学研究科

毛利 公一 ††
†† 立命館大学情報理工学部

1 はじめに

近年，仮想化技術を搭載したマルチコアプロセッサが登場し，普及してきている．マルチコアプロセッサを搭載した計算機は，シングルプロセッサの計算機と比べ，並列処理を利用したプロセスの実行が優位である．また，1 台の計算機上に複数の OS を起動させることが可能となっており，このように提供されている仮想計算機 (以下 VM と記す) には，物理的な計算機の資源を自由に割り振ることが可能となっている．近年，高性能化している計算機資源では，今までの計算機 1 台に対して OS1 つといった利用方法と比べ，仮想化技術を用いて 1 台の計算機上で複数の OS を起動させることでより柔軟な資源利用が可能となる．

このようにマルチコアプロセッサと仮想化技術を組み合わせて利用することで，複数の VM に任意の物理的なコアを割り当てることが可能となり，それぞれの VM の処理能力を指定することが可能となった．これにより，従来，計算機に依存していた OS の処理能力を自由に変更することが可能となり，OS の特性によってプロセッサの割当てを指定することで，計算機資源を有効に利用できるようになった．

従来，仮想計算機モニタ (以下，VMM と記す) と OS は，独自に最適な資源配分を行っており，VMM は，物理 CPU(以下，PCPU と記す) を仮想 CPU(以下，VCPU と記す) に割当て，OS は CPU をプロセスへ割当てている．しかし，ある VM に割当てられている VCPU が利用されておらず，別の VM に割り振られている VCPU の処理能力が不足しているような場合，システム全体としては効率的な計算機資源の割付けができているとはいえない．そこで，VMM が，VM 上で起動している OS と動作状況に関する情報を交換し，適応的に資源を管理・提供することによって，各 OS が持つ特性に従った要求を満たすことが可能とする．また，このような VMM が動的に資源を提供するような環境において，VM 上で動作する OS は動的に提供される資源を識別し，利用する必要がある．

本稿では，このような，VMM と OS とが協調し動作する協調型仮想計算機システムの構築を行うに際し，必要な協調機構について述べる．

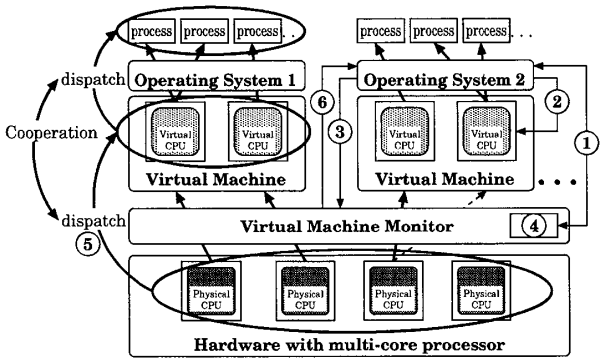

図 1 協調型仮想計算機システム

2 協調型仮想計算機システム

2.1 VMM と OS の協調

協調型仮想計算機システムでの，VMM とゲスト OS との協調とは，お互いに情報交換を行いながら効率的な資源利用を行うことであり，本稿では特に，物理的に有限である資源のプロセッサに着目する．

図 1 に示すように，VMM では，PCPU と VCPU との割当てをスケジューリングしており，これによってゲスト OS に資源を提供する．また，ゲスト OS は，提供された VCPU のスケジューリングを行い，プロセスに資源を割当てる．

2.2 協調型スケジューリング

資源スケジューリングを，協調して行うため，VMM と OS との間で通信する機構を実装する．相互にプロセッサ利用率や，スケジューリング対象となっているプロセスといった情報を交換することで，VMM と各 OS 間での総合的なスケューリングを行う．

各 OS は自身のプロセスの要求，現在の状況などに従い，VMM へと VCPU の増減要求を発行する．OS からの VCPU 増減要求を受信した VMM は，全 VM 上で動作している OS の状況を考慮し，VM の資源を管理する．その結果，VM 上の VCPU の増加があった場合，VMM は対象の OS に対し VCPU の増加を通知する．VMM からの通知を受理した OS は VCPU の確保を行い，自身のプロセスを確保している VCPU に対してスケジューリングを行う．また，VMM から VCPU の解放要求があった場合，OS は自身が確保している VCPU を開放

A cooperative mechanism for virtual machine monitor and guest OS
Hiroyasu Araki†, and Koichi Mouri††
†Graduate School of Science and Engineering, Ritsumeikan University
††College of Information Science and Engineering, Ritsumeikan University

し，VMMは開放されたVCPUに対するPCPUの割当てを削除する．

このような協調型スケジューリングを行う場合，OSとVMMに，資源の増減に対応した機構を実装するだけでなく，双方向の通信機構を備える必要がある．これらの機構を実現することにより，VMMとOSの連携の取れたスケジューリングが可能であり，システム全体でのにパフォーマンスの向上が見込まれる．

3 協調機構

協調型仮想計算機システムを構築するにあたりVMMとOSに必要な機構を図1中に示し，それぞれについて以下で述べる．まず，ゲストOSに必要な機構を以下に示す．

① VMMとの通信機構
OS自身が，プロセッサ利用率，プロセス特性等の情報をVMMに対して送信する．これによりVMMは各OSの動作状況を詳しく知ることができる．また，VMMからVCPUの割当て状況を受信することで，VCPUをスケジューリング対象から削除し停止状態を実現する．

② VM上の資源管理機構
VMMによって割当てが変動されるVMの動的な資源の変動に対応する．新たに提供されたVCPUの利用や，VCPUの利用停止を実現する．

③ VMMへの資源変動要求機構
資源の変動要求をVMMに通知することで，OSからVMMに対して資源の増加，減少要求の発行を実現する．

次に，VMMに必要な機構を以下に示す．

④ OSの情報管理機構
OSとの通信によって得られた情報を管理し，システム全体の利用状況を把握する．OSから資源要求があった場合にシステム全体を考慮したPCPUの割当てが可能となる．また，各OSの不足，過剰資源の認知を実現する．

⑤ VMの資源変動機構
VM上に割り振られている資源を管理し，実際にVMの資源を変動させる．新たなVCPUの構築，VCPUへのPCPUの割当て，解除を実現する．

⑥ 要求に応答する機構
OSに対してVCPUの割当て状況，新規に構築したVCPUの情報などを通知する．これによって，OSは自身が出した要求にVMMがどう対応したかを認知することが可能となり，それらに対応した処理を行うことが可能となる．

4 実装状況

現在，VMMのターゲットとしてXen[1]，ゲストOSのターゲットとしてLavender[2]を用いている．Xenは準仮想化と完全仮想化の二つの仮想化環境を提供しており，本研究では，協調機構の実装の必要性から準仮想化環境を用いることとする．そこで，協調機構を実装するにあたり，まず，Lavenderの準仮想化に取り組んでいる．準仮想化に対応するためには，ゲストOSに変更を加える必要があり変更点は以下のようなものがある．

1. Xenから提供される情報の利用
2. ハイパーバイザコールの利用

Xenは，VMの構築時に提供するVMの情報を，Xenとゲスト OSの両方が参照可能な領域である，共有メモリ領域に用意する．ゲストOSは起動時にこれを元に起動し，起動後もハードウェア情報の取得などにこれを利用する．まずは，この共有メモリ領域の情報を元にした起動を行うような変更を加える．

ゲストOSが，割り込みの初期化やメモリ管理といった，特権命令を利用し，ハードウェアの操作を行うような処理をする際，準仮想化環境ではハイパーバイザコールが利用される．そこで，ゲストOS内の処理で，このような特権命令を利用する一連の処理をハイパーバイザコールに置き換える必要がある．

現在，Lavenderにおいて，上記1で記す，準仮想化での起動の実装が完了しており，ハイパーバイザコールの利用についての実装を順次行っている．

5 おわりに

本稿では，独立した資源管理を行っていたVMMとOSが，それぞれ動作状況を共有し協調することで，より効率的な資源利用が可能となる協調型仮想計算機システムを提案した．また，このシステムを構築するために必要である機構をVMM，OSのそれぞれについて述べた．これらの機構をVMM，OSに実装することで協調したスケジューリングが可能となる．現在，3章で述べた機構の実装を行うため，Lavenderの準仮想化に取り組んでいる．この作業が完了しだい，順次，協調機構を実装し，協調型仮想計算機システムの構築を行う．

参考文献

[1] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield:"Xen and the Art of Virtualization", In Proceedings of the 19th ACM Symposium on Operating Systems Principles, pp.164-177 (2003).

[2] 毛利公一，大久保英嗣:"マイクロカーネルLavenderの設計と開発"，電子情報通信学会論文集 (D-I)，Vol.J82-D-I，No. 6，pp. 730-739 (1999).